**Assignment No: - 6**

**Name :- Isha Ghorpade**

**Enrollnment :- 22420020**

**Roll :- 381066**

**Problem Statement:**

Sentiment analysis using LSTM network or GRU.

**Objectives**

* Preprocess and clean textual data for NLP tasks.
* Build and train LSTM/GRU-based models for sentiment classification.
* Evaluate performance using accuracy, confusion matrix, and other metrics.
* Compare efficiency of LSTM and GRU models for sentiment analysis.

**Software Packages and Hardware Apparatus**

Software Packages

* Python → Primary programming language for NLP and deep learning.
* Jupyter Notebook → Interactive environment for coding and visualization.
* TensorFlow/Keras → Frameworks for building and training neural networks.
* NLTK → Natural language processing library for stopword removal and text cleaning.
* Scikit-Learn → For data splitting, evaluation, and confusion matrix.

Hardware Apparatus

* System with at least 8 GB RAM for smooth training and testing.
* GPU (recommended) for faster model training.
* CPU (usable if GPU unavailable, but slower training).

**Libraries Used**

* Pandas → Load and manipulate dataset (IMDB reviews).
* NumPy → Numerical computations and reshaping.
* NLTK → Stopword removal and text cleaning.
* Scikit-Learn → Dataset splitting and evaluation metrics.
* TensorFlow/Keras →
  + Sequential → Define model architecture.
  + Embedding → Create word embeddings.
  + LSTM / GRU → Capture sequential dependencies.
  + Dense → Fully connected output layer.
  + ModelCheckpoint → Save best model.
* re (Regex) → Remove HTML tags, special characters, and clean text.

**Theory**

1. Sentiment Analysis

A branch of NLP used to determine emotional polarity (positive, negative, neutral) in text. Applications include customer reviews, social media, healthcare, and finance.

2. LSTM (Long Short-Term Memory)

A special type of RNN designed to capture long-term dependencies in sequential data. It uses memory cells and gates:

* Forget Gate → Decides what information to discard.
* Input Gate → Adds new information to memory.
* Output Gate → Produces results based on memory.

3. GRU (Gated Recurrent Unit)

A simpler version of LSTM that merges the forget and input gates into an update gate. Advantages:

* Fewer parameters → Faster training.
* Similar performance to LSTM in many tasks.

**Methodology**

1. Data Loading
   * Load IMDB dataset containing labeled reviews (positive/negative).
2. Data Cleaning & Preprocessing
   * Remove HTML tags and special characters using regex.
   * Convert text to lowercase.
   * Remove stopwords using NLTK.
3. Sentiment Encoding
   * Encode target variable: Positive = 1, Negative = 0.
4. Data Splitting
   * Split dataset into 80% training, 20% testing.
5. Tokenization & Padding
   * Use Keras Tokenizer to convert text into integer sequences.
   * Apply padding/truncation to ensure fixed sequence length.
6. Model Building
   * Define Sequential model.
   * Add Embedding layer for word vectors.
   * Add LSTM or GRU layer to capture dependencies.
   * Add Dense output layer with sigmoid activation.
7. Model Training
   * Compile model with binary\_crossentropy loss and Adam optimizer.
   * Train with batch size (e.g., 32) and epochs (e.g., 5–10).
   * Use ModelCheckpoint to save best performing model.
8. Model Evaluation
   * Evaluate on test dataset (accuracy, loss).
   * Use confusion matrix to assess classification performance.

**Advantages**

* Captures sequential patterns in text effectively.
* Robust preprocessing improves noise tolerance.
* Adaptable to multiple languages with minimal adjustments.
* High accuracy possible with sufficient data and tuning.

**Limitations**

* Computationally expensive → Needs GPU for efficiency.
* Training is time-consuming for large datasets.
* Data sensitive → Preprocessing and hyperparameters heavily impact performance.
* Overfitting risk → Requires dropout/regularization.

**Applications**

* Customer Review Analysis → Product & service feedback classification.
* Social Media Monitoring → Tracking sentiment towards brands/events.
* Healthcare → Patient review analysis for better services.
* Financial Markets → Sentiment from news to predict stock trends.

**Working / Algorithm**

Step 1: Import libraries.  
Step 2: Load IMDB dataset.  
Step 3: Preprocess text → clean, lowercase, remove stopwords.  
Step 4: Encode labels (positive=1, negative=0).  
Step 5: Split into train/test (80/20).  
Step 6: Tokenize words & pad sequences.  
Step 7: Build model:

* Embedding → LSTM/GRU → Dense(sigmoid).  
  Step 8: Compile & train (Adam optimizer, binary\_crossentropy).  
  Step 9: Evaluate accuracy on test data.  
  Step 10: Save model for deployment.

**Conclusion**

This practical demonstrates sentiment analysis using LSTM and GRU networks. Both models effectively capture sequential dependencies in text and achieve good accuracy when trained on IMDB reviews. While LSTM provides strong performance for long dependencies, GRU offers computational efficiency with similar results.

With proper preprocessing, hyperparameter tuning, and regularization, these models achieve high accuracy in real-world sentiment analysis tasks, making them useful for customer review mining, social media monitoring, healthcare feedback analysis, and financial forecasting.